

# Прости пресмятания

Работа с конзола, аритметични  
операции с числа



**СофтУни**

трейнерски екип

Софтуерен университет

<http://softuni.bg>



**Основи на  
програмирането**

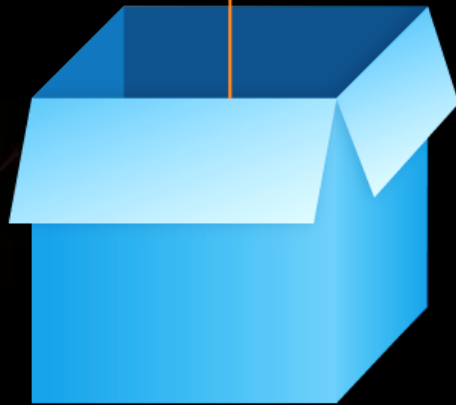


# Съдържание

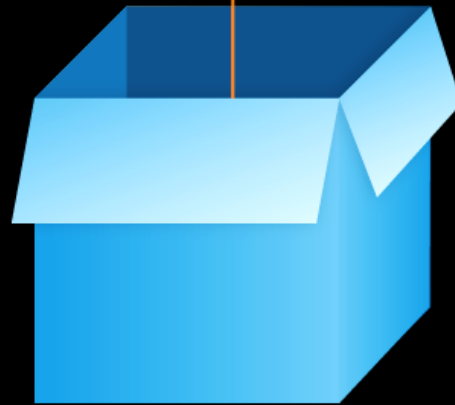
1. Променливи и типове данни
2. Четене на потребителски вход
3. Прости операции
  - Работа с текст
  - Работа с числа
4. Печатане на екрана



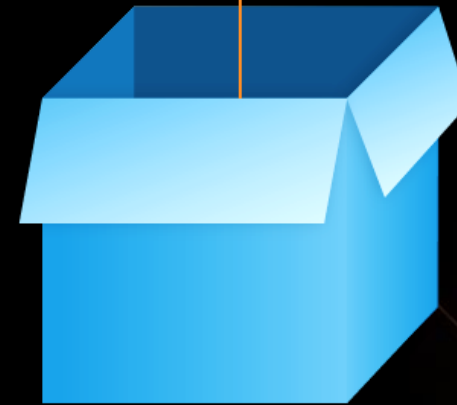
"Bob"



true



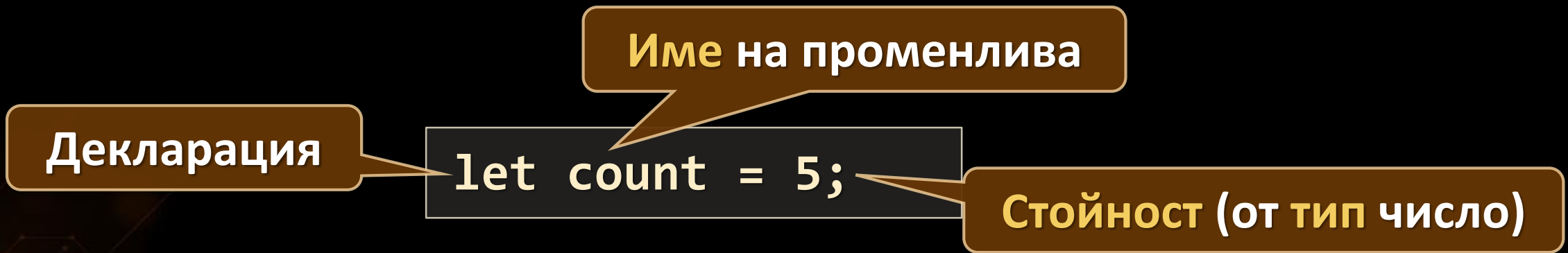
35



**Променливи и типове данни**

# Променливи

- Компютрите са машини, които обработват данни
  - Данните се записват в компютърната памет в **променливи**
  - Променливите имат **име, тип и стойност**
- Дефиниране на променлива и присвояване на стойност:



- След обработка данните се записват отново в променливи

# Типове данни

- Променливите съхраняват стойност от даден тип
  - Число, текст(низ), булев тип, списък, ...
- Типове данни – примери:
  - number - число: **7**, **3.14**, **-1.5**, **NaN**, ...
  - string - текст : **"Hello"**, **"Здрасти"**, **"p@r0La"** ...
  - boolean - булева: **true**, **false**
  - Date - дата: **Tue Jul 04 2017**, ...
  - null: **null**
  - undefined: **undefined**



**Четене на потребителски вход**

# Четене на текст

- Четене на текст:

```
function readText([input]) {  
  let str = input;  
}
```

- Пример:

```
function readName([input]) {  
  let name = input;  
  console.log(name);  
}
```

# Четене на числа

- Четене на число:

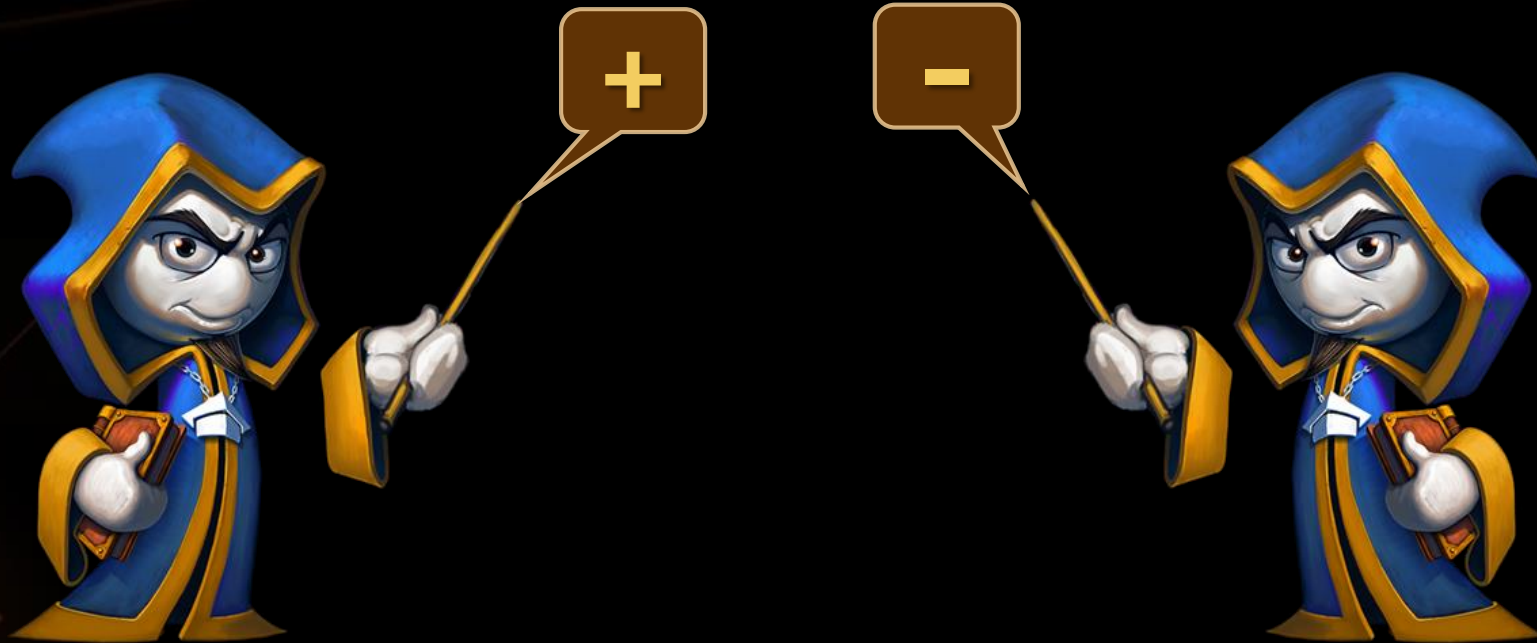
```
function readNumber([input]) {  
  let num = Number(input);  
}
```

- Пример: пресмятане на лице на квадрат със страна **a**:

```
function squareArea([input]) {  
  let a = Number(input);  
  let area = a * a;  
  console.log(area);  
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/151#0>





# Прости операции

## Работа с текст и числа

# Поздрав по име - пример

- Да се напише програма, която:
  - чете **име** на човек, въведено от потребителя
  - отпечатва "**Hello, <name>!**", където **<name>** е въведеното преди това име.

# Поздрав по име - решение

Операцията "+" долепя

```
function greetingByName([input]) текстовете  
  let name = input;  
  let greeting = "Hello, " + name;  
  console.log(greeting);  
}
```

# Съединяване на текст и число

- Съединяване на текст и число (оператор +):

```
let firstName = "Maria";  
let lastName = "Ivanova";  
let age = 19;  
let str = firstName + " " + lastName + " @ " + age;  
console.log(str); // Maria Ivanova @ 19
```

Резултатът е  
долепяне/конкатенация

```
let a = 1.5;  
let b = 2.5;  
let sum = "The sum is: " + a + b;  
console.log(sum); // The sum is 1.52.5
```

# Аритметични операции: + и -

- Събиране на числа (оператор +):

```
let a = 5;  
let b = 7;  
let sum = a + b; // 12
```

- Изваждане на числа (оператор -):

```
function subtract([arg1, arg2]) {  
  let a = Number(arg1);  
  let b = Number(arg2);  
  let result = a - b;  
  console.log(result);  
}
```

# Аритметични операции: \* и /

- Умножение на числа (оператор \*):

```
let a = 5;  
let b = 7;  
let product = a * b; // 35
```

- Деление на числа (оператор /):

```
let a = 25;  
let i = a / 4; // 6.25 (дробно делене)  
let f = parseInt(a / 4.0); // 6 (дробната част се отрязва)  
let infinity = a / 0; // 6 получава се Infinity (безкрайност)  
let sqrt = Math.sqrt(-1); // получава се NaN (не число)
```

# Числени изрази

- В програмирането можем да пресмятаме числени изрази

```
let expr = (3 + 5) * (4 - 2);
```

- Изчисляване на **лице на трапец**

```
function trapezoidArea([arg1, arg2, arg3]) {  
  let b1 = Number(arg1);  
  let b2 = Number(arg2);  
  let h = Number(arg3);  
  let area = (b1 + b2) * h / 2;  
  console.log("Trapezoid area = " + area);  
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/151#4>

```
C:\Program Files\nodejs\node.exe  
Debugger listening on [::]:5858  
centimeters = 12.7
```

```
C:\Program Files\nodejs\node.exe  
Debugger listening on [::]:5858  
Triangle area = 262.50
```

```
C:\Program Files\nodejs\node.exe  
Debugger listening on [::]:5858  
Trapezoid area = 73.5
```

```
C:\Program Files\nodejs\node.exe  
Debugger listening on [::]:5858  
Date after 1000 days is: 26 9 2014
```

```
C:\Program Files\nodejs\node.exe  
Debugger listening on [::]:5858  
BGN = 35.91
```

```
C:\Program Files\nodejs\node.exe  
Debugger listening on [::]:5858  
You are Ivan Ivanov, a 25-years old person from Burgas.
```

# Задачи с прости изчисления

## Работа на живо в клас (лаб)



# Особености при операции с числа в JavaScript

- При събиране и изваждане на числа, ако не превърнем стойността на променливата в число:

```
let a = 15;  
console.log(a / 2.0); // Дробен резултат: 7.5  
console.log(a / 0.0); // Резултат: Infinity  
console.log(0.0 / 0.0); // Резултат: NaN
```

- При умножение и деление на числа се получават отклонения в остатъка:

```
let b = 23 * 1.3; // 29.900000000000000002
```

```
b.toFixed(2) = "29.90"
```

IEEE 754 – стандарт за представяне на числата с плаваща запетая



```
`${...}`
```



**Печатане на екрана**  
Форматиране на изход

# Съединяване на текст

- При печат на текст, числа и други данни, можем да ги съединим, използвайки интерполация **`${arg1}`**, **`${arg2}`**, **`${arg3}`** :
- Пример: поздрав по име:

Интерполация на стринг се извършва с **`back-tick(`)`**

```
function greet([name]) {  
  console.log(`Hello, ${name}!`);  
}
```

Изразът **`${name}`** се замества с аргумента **`name`**

Hint: Template literals

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/151#2>

# Съединяване на текст и числа - пример

- Да се напише програма, която:
  - чете **име, фамилия, възраст** и **град**, въведени от потребителя
  - отпечатва "You are **<firstName> <lastName>**, а **<age>-years old person from <town>**".

# Съединяване на текст и числа - решение

```
function personalInfo([arg1, arg2, arg3, arg4]) {  
  let firstName = arg1;  
  let lastName = arg2;  
  let age = Number(arg3);  
  let town = arg4;  
  console.log(`You are ${firstName} ${lastName}, a  
  ${age}-years old person from ${town}.`);  
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/151#3>

# Закръгляне на числа

- В програмирането можем да закръгляме дробни числа

- Закръгляне до следващо (по-голямо) цяло число:

```
let up = Math.ceil(23.45); // up = 24
```

- Закръгляне до предишно (по-малко) цяло число:

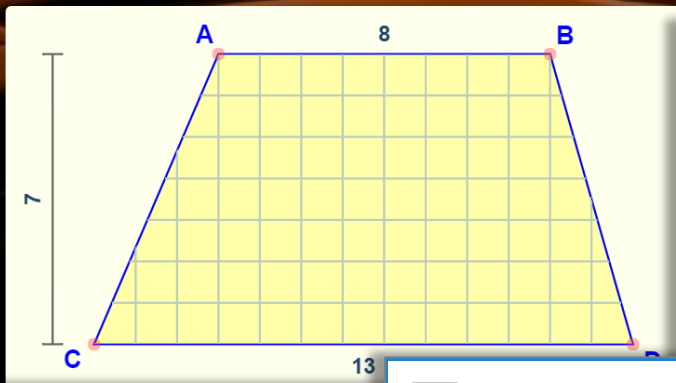
```
let down = Math.floor(45.67); // down = 45
```

- Отрязване на знаците след десетичната запетая:

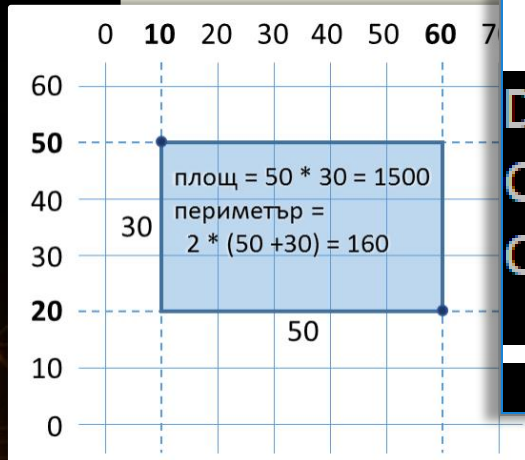
```
let trunc = Math.trunc(45.67); // trunc = 45
```

- Закръгляне до най-близко число.

```
(123.456).toFixed(2); // 123.46
```



```
C:\Program Files\nodejs\node.exe  
Debugger listening on [::]:5858  
Trapezoid area = 73.5
```



```
C:\Program Files\nodejs\node.exe  
Debugger listening on [::]:5858  
Circle area = 63.62  
Circle perimeter = 28.27
```

# Лице и периметър на фигури

Работа на живо в клас (лаб)

# Периметър и лице на кръг – пример

- Напишете програма, в която потребителят въвежда радиуса **r** на кръг и изчислява лицето и периметъра на кръга / окръжността
  - Лице =  $\pi * r * r$
  - Периметър =  $2 * \pi * r$

$$\pi \approx 3.14159265358979323846\dots$$



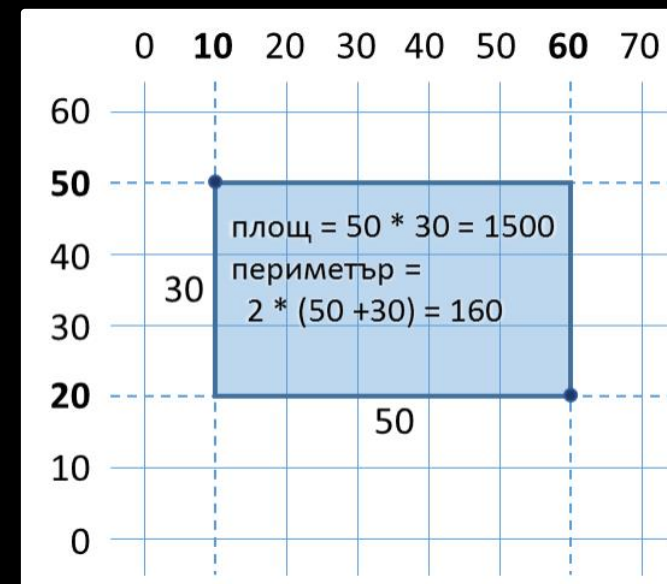
# Периметър и лице на кръг – решение

```
function circleAreaPerimeter([arg1]) {  
  let r = Number(arg1);  
  console.log(Math.PI * r * r);  
  console.log(2 * Math.PI * r);  
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/151#5>

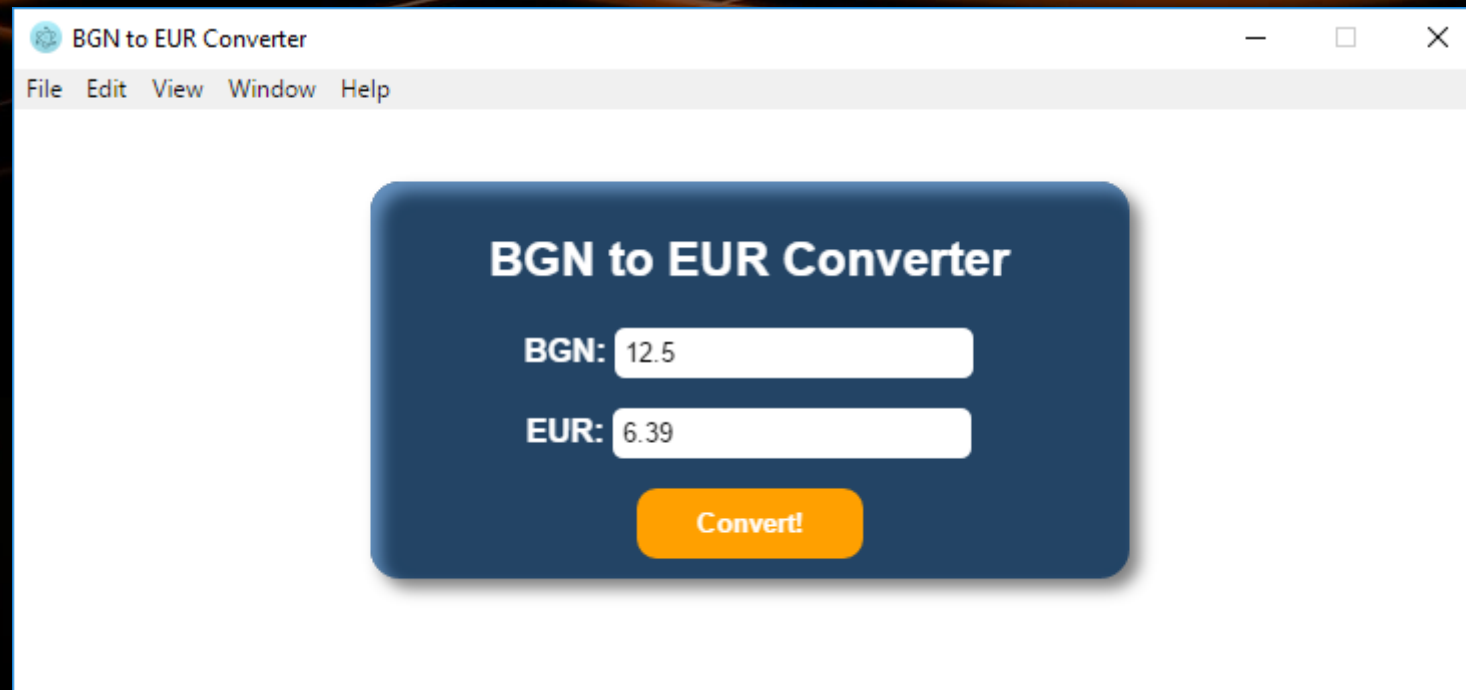
# Лице на правоъгълник – пример

- **Правоъгълник** е зададен с координатите на два от своите срещуположни ъгъла
  - Да се пресметнат **площта** и **периметъра** му



# Лице на правоъгълник – решение

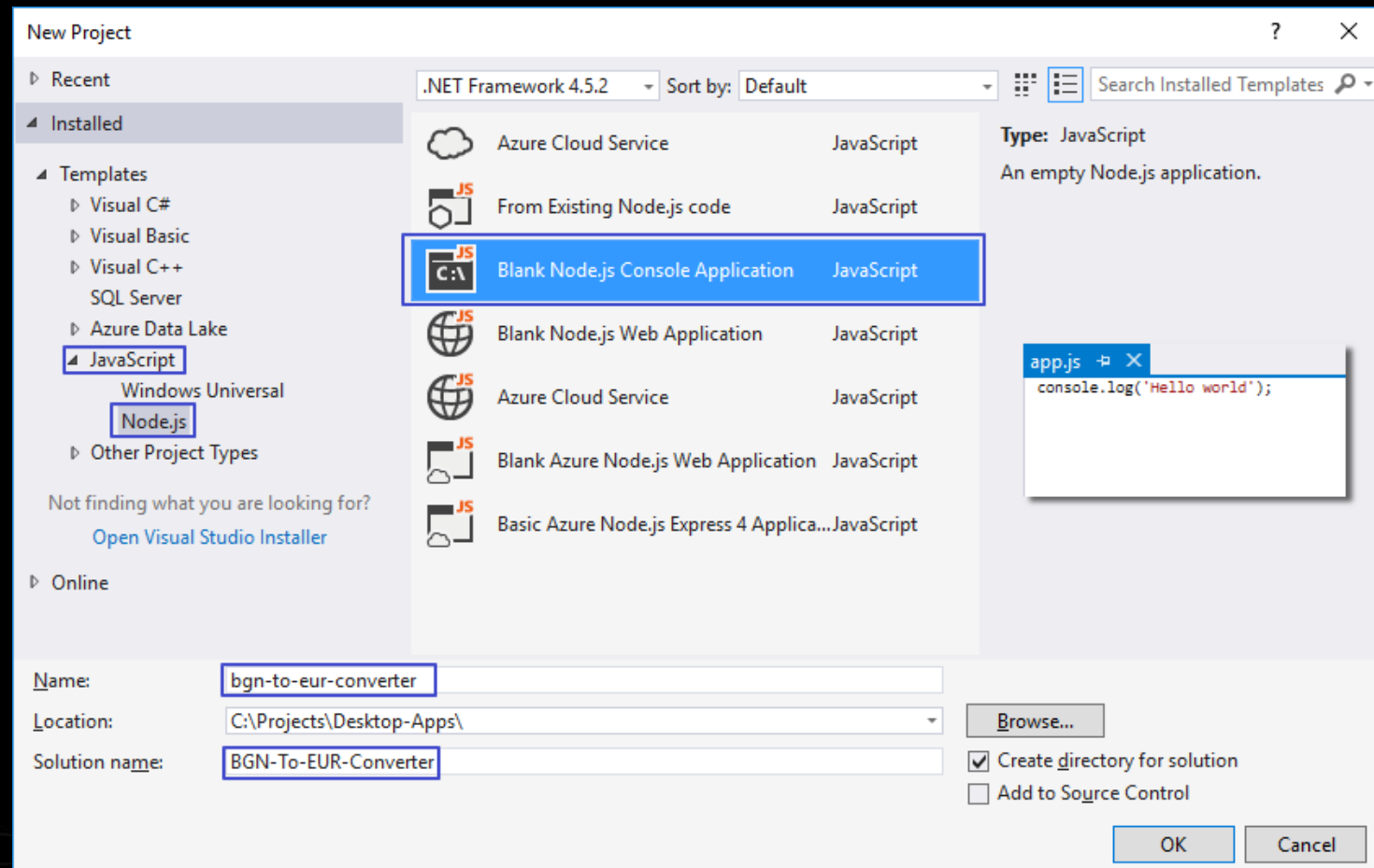
```
function rectAreaPerimeter([arg1, arg2, arg3, arg4]) {  
  let x1 = Number(arg1);  
  let y1 = Number(arg2);  
  let x2 = Number(arg3);  
  let y2 = Number(arg4);  
  
  let width = Math.max(x1, x2) - Math.min(x1, x2);  
  let height = Math.max(y1, y2) - Math.min(y1, y2);  
  
  console.log(`Area = ${width * height}`);  
  console.log(`Perimeter = ${2 * (width + height)}`);  
}
```



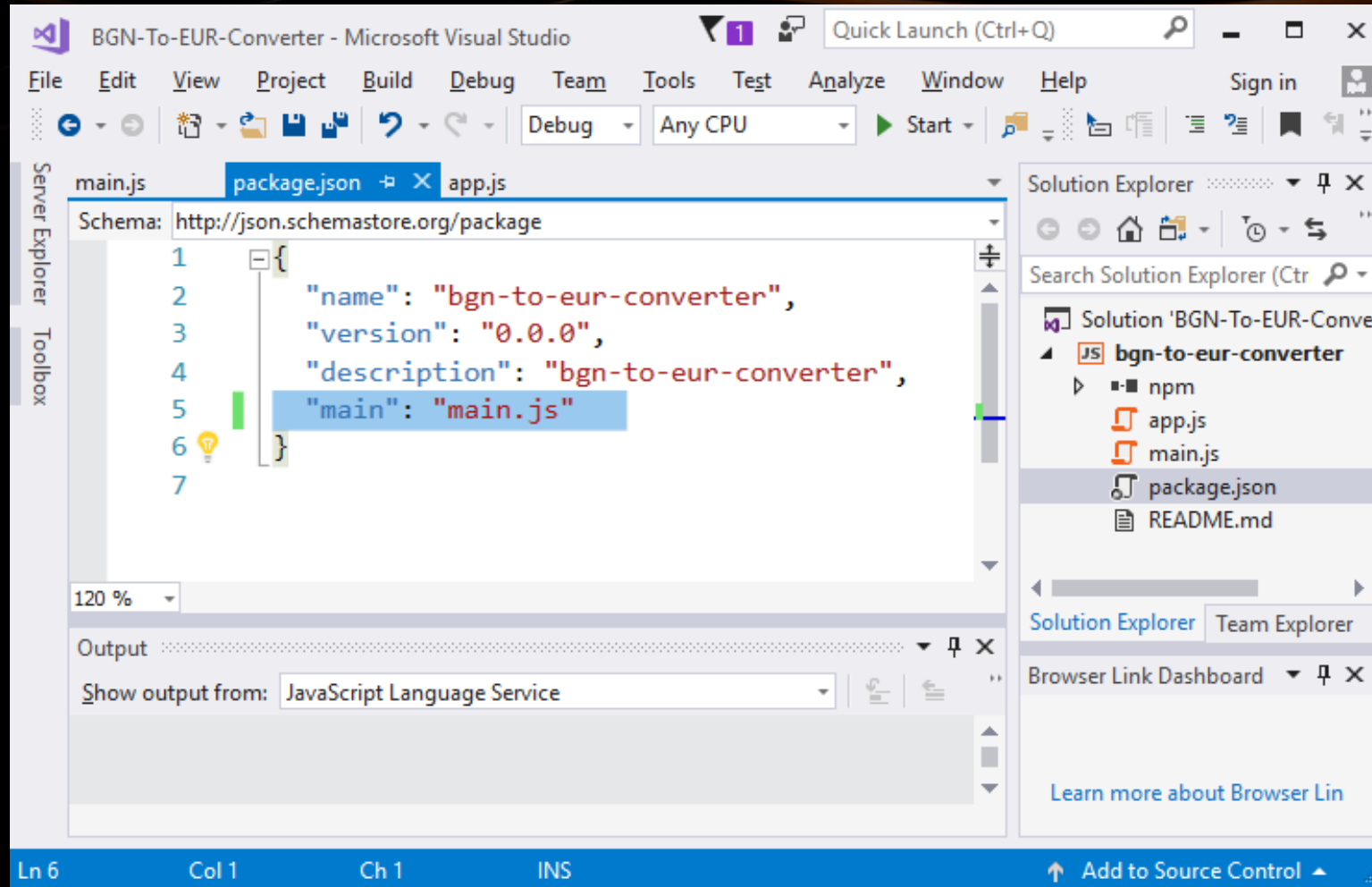
**Графично приложение:  
конвертор от BGN към EUR**

# Конвертор за валути

- Създаване на ново конзолно приложение Node.js



# Структуриране на проекта



The screenshot displays the Microsoft Visual Studio interface for a project named "BGN-To-EUR-Converter". The main editor window shows the `package.json` file with the following content:

```
1  {  
2    "name": "bgn-to-eur-converter",  
3    "version": "0.0.0",  
4    "description": "bgn-to-eur-converter",  
5    "main": "main.js"  
6  }  
7
```

The `"main": "main.js"` line is highlighted in blue. The Solution Explorer on the right shows the project structure:

- Solution 'BGN-To-EUR-Conve'
  - JS bgn-to-eur-converter
    - npm
    - app.js
    - main.js
    - package.json
    - README.md

The Output window at the bottom shows the JavaScript Language Service. The status bar at the bottom indicates the current cursor position: Ln 6, Col 1, Ch 1, INS.

# Подготвяне на html файла

```
<body>
  <form class="content-form">
    <h2 class="title">BGN to EUR Converter</h2>
    <section class="items">
      <label for="bgn" class="currency">
        <span class="item-currency">BGN: </span>
        <input class="currency-value" type="number" id="bgn" value="0" />
      </label>
      <label for="euro" class="currency">
        <span class="item-currency">EUR: </span>
        <input class="currency-value" type="text" id="euro" readonly />
      </label>
      <input class="primary-btn" type="button" value="Convert!" />
    </section>
  </form>
</body>
```

# Инсталиране на Electron.js

```
C:\WINDOWS\system32\cmd.exe

C:\Projects\DesktopApps>npm install electron -g
```



```
C:\WINDOWS\system32\cmd.exe

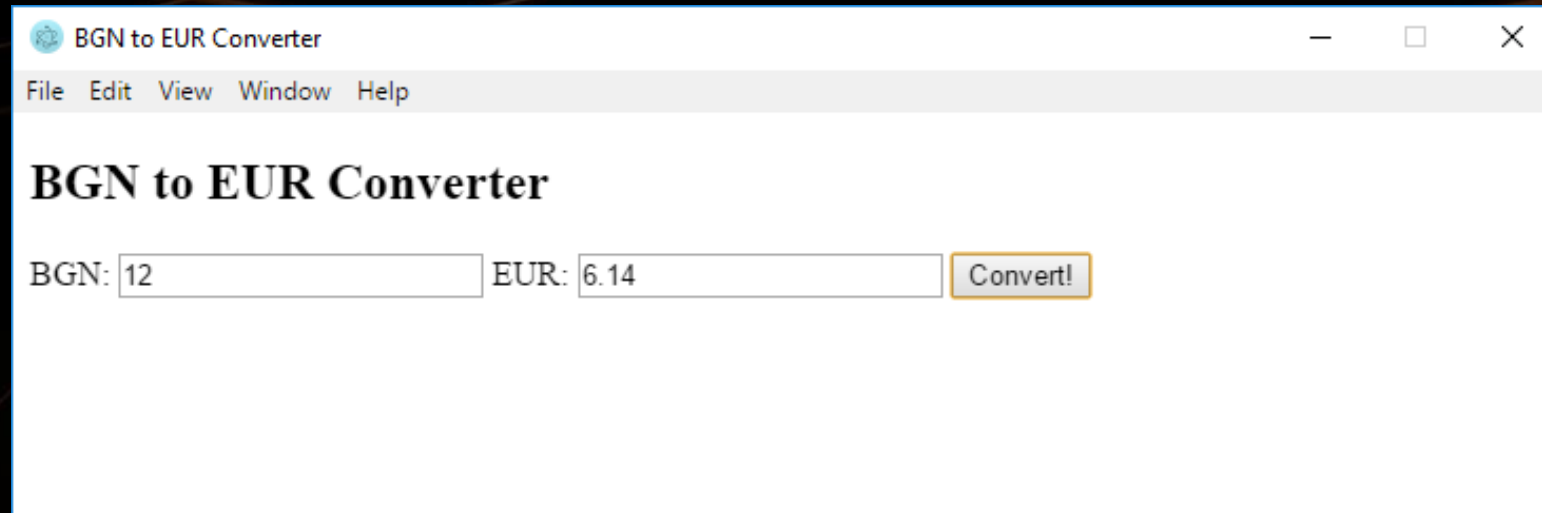
| `-- typedarray@0.0.6
+-- debug@2.2.0
| `-- ms@0.7.1
+-- mkdirp@0.5.0
| `-- minimist@0.0.8
`-- yauzl@2.4.1
    `-- fd-slicer@1.0.1
        `-- pend@1.2.0

C:\Projects\DesktopApps>_
```



# Конвертиране от BGN към EUR – логика

```
function eurConverter() {  
    let bgn = document.getElementById("bgn").value;  
    let eur = (bgn / 1.95583).toFixed(2);  
    document.getElementById("euro").value = eur;  
}
```



**Графично приложение:  
конвертор от BGN към EUR**

**Работа на живо в клас (лаб)**

# Какво научихме днес?

- Въвеждане на текст

```
let str = arg1;
```

- Въвеждане на число

```
let num = Number(arg1);
```

- Пресмятания с числа: +, -, \*, /, ()

```
let sum = 5 + 3;
```

- Извеждане на текст по шаблон

```
console.log(` ${arg1} + ${arg2} = ${arg1 + arg2} `);
```



# Прости пресмятания



## Въпроси?



- Настоящият курс (слайдове, примери, видео, задачи и др.) се разпространяват под свободен лиценз "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International"



# Безплатни обучения в СофтУни

- Фондация "Софтуерен университет" – [softuni.org](http://softuni.org)
- Софтуерен университет – качествено образование, професия и работа за софтуерни инженери
  - [softuni.bg](http://softuni.bg)
- СофтУни @ Facebook
  - [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)
- СофтУни форуми – [forum.softuni.bg](http://forum.softuni.bg)

